ELSEVIER

Discrete Applied Mathematics 117 (2002) 65–79

DISCRETE APPLIED MATHEMATICS

Complexity of list coloring problems with a fixed total number of colors

Sylvain Gravier[a,b,*,1], Daniel Kobler[c,2], Wieslaw Kubiak[d,e,3]

[a] *CNRS, Laboratoire Leibniz-IMAG, 46 avenue Félix Viallet, 38031 Grenoble Cedex 1, France*
[b] *Eötvös University, Múzeum kőrút 6-8, Budapest H-1088, Hungary*
[c] *École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*
[d] *Laboratoire GILCO, École Nationale Supérieure de Génie Industriel, France*
[e] *Laboratoire Leibniz, Université Joseph Fourier, France*

Received 5 January 1999; received in revised form 28 November 2000; accepted 11 December 2000

Abstract

We study list coloring problems where the total number k of colors on all lists is fixed. Such problems are known to be $\mathcal{NP}$-Complete even for planar bipartite graphs and $k=3$. We give polynomial algorithms for some special cases of these problems.

Keywords: Complexity; List coloring

1. Introduction

We will consider undirected, finite, simple graphs. A coloring of a graph $G=(V,E)$ is a mapping $c: V \to \mathbb{N}$ such that $c(v_i) \neq c(v_j)$ for every edge $(v_i, v_j) \in E$. A coloring which uses at most k colors is called a k-coloring. Each color class is a stable set, hence a k-coloring can be seen as a partition of V into stable sets $S_1, \ldots, S_k$. A graph is called k-colorable if it admits a k-coloring. Deciding whether there exists a k-coloring for a given graph G and a given integer k is known to be $\mathcal{NP}$-complete [11] even for $k=3$. The following extension of this problem has been proposed by Vizing [22] and Erdős et al. [9], the *list coloring problem* denoted by (G,L):

* Corresponding author. CNRS, Laboratoire Leibniz-IMAG, 46 Avenue Felix Viallet, F-38031 Grenoble Cedex 1, France.

E-mail addresses: sylvain.gravier@imag.fr (S. Gravier), kobler@dma.epfl.ch (D. Kobler), wieslaw.kubiak@imag.fr (W. Kubiak).

[1] Research supported by a TEMPRA program (France) under grant GLM6C001.

[2] Research supported by the Swiss National Fund for Scientific Research under grant 21-45070.95.

[3] On leave from the Faculty of Business Administration, Memorial University of Newfoundland St. John's, Nfld, Canada. Research supported by the Natural Sciences and Engineering Research Council of Canada under Grant OGP0105675, and by the KBN of Poland under Grant 8T11A01618.

0166-218X/02/$ - see front matter
PII: S0166-218X(01)00179-2

Instance: A graph $G=(V,E)$ and for each vertex $v\in V$, a list $L(v)$ of colors allowed for v.

Question: Is there a coloring c of G such that $c(v)\in L(v), v\in V$?

If such a coloring c exists, then we call c an L-coloring of G, and we say that G is L-colorable. This problem is $\mathcal{NP}$-complete even for interval graphs [1], for complete bipartite graphs [13], and for line-graphs of complete bipartite graphs [10]. However, (G,L) is solvable in polynomial time for partial t-trees with fixed t (in $O(|V|\cdot k^{t+1}\cdot t)$ time) [16]. If the number of available colors $k=|L(V)|=|\bigcup_{v\in V}L(v)|$ is fixed, then (G,L) (denoted by k-(G,L)) is also solvable in polynomial time for P_4-free graphs (in $O((|V|+|E|)\cdot 4^k)$ time) [16]. de Werra [23] introduced the *list coloring problem with cardinalities* denoted by (G,L,p):

Instance: A graph $G=(V,E)$, a list $L(v)$ of colors allowed for vertex v, and a mapping p which associates a positive integer $p(j)$ to each color $j\in L(V)$ (p is called a color-mapping).

Question: Is there a coloring c of G such that $c(v)\in L(v)$, $v\in V$ and such that $|c^{-1}(j)|=p(j)$, $j\in L(V)$?

Thus, the goal is to find a coloring of the graph such that each vertex is assigned a color from its list, and such that there are exactly $p(j)$ vertices of color j. Without loss of generality, we may suppose that $L(v)\subseteq\{1,\dots,k\}$ for each vertex $v\in V$, with $k=|L(V)|$. We will represent by k-(G,L,p) the corresponding restricted list coloring problem with fixed number of available colors $k=|L(V)|$. de Werra [23] proved that (G,L,p) is polynomial for G being a union of disjoint cliques. When $G=P_n$ is a path with $|V|=n$ vertices, de Werra conjectured that (P_n,L,p) is $\mathcal{NP}$-complete. The proof of this conjecture is given by Dror et al. [8]. The authors in [8] proved a stronger result, namely that (P_n,L,p) is $\mathcal{NP}$-complete even if $|L(v)|\leqslant 2$ for each vertex $v\in V$. However, the problem k-(P_n,L,p) can be solved in polynomial time ($O(n^k)$) by dynamic programming [8].

In the next section, we will show that k-(G,L) reduces to k-(G,L,p), which means that if there exists a polynomial time algorithm for k-(G,L,p), then there exists a polynomial time algorithm for k-(G,L). This reduction, and the proof in [18] demostrating that 3-(G,L) is $\mathcal{NP}$-Complete even for a planar bipartite graph G, show that 3-(G,L,p) is also $\mathcal{NP}$-Complete even when G is a bipartite planar graph. We give a polynomial time algorithm for 2-(G,L,p). In Sections 3 and 4, we give a polynomial time algorithm for k-(G,L,p) (and hence k-(G,L)) for two classes of graphs containing P_4-free graphs and triangulated graphs. These two results extend some earlier results given from [16,8].

2. Complexity results

2.1. Complexity of k-(G,L,p) for $k\geqslant 3$

To show that k-(G,L) can be considered an instance of k-(G,L,p), we introduce the *list coloring problem with bounded cardinalities* denoted by k-$(G,L,\leqslant p)$:

Instance: A graph $G=(V,E)$, a list $L(v)$ of colors allowed for each vertex $v\in V$ such that $L(V)\subseteq\{1,\ldots,k\}$, and a color-mapping p.

Question: Is there a coloring c of G such that $c(v)\in L(v), v\in V$ and such that $|c^{-1}(j)|\leqslant p(j)$ for every color $j\in\{1,\ldots,k\}$?

Lemma 1. *k-$(G,L,\leqslant p)$ and k-(G',L',p) are polynomially equivalent.*

Proof. The problem k-(G',L',p) is a special case of k-$(G,L,\leqslant p)$ with $\sum_{i\in L(V)}p(i)=n$, $G=G'$, $L=L'$ and n being the number of vertices of G'. Now, let a graph $G=(V,E)$, a list $L(v)$ of allowed colors for each $v\in V$ such that $L(V)\subseteq\{1,\ldots,k\}$, and a color-mapping p constitute an instance of k-$(G,L,\leqslant p)$. We may assume that $t=\sum_{i\in L(V)}p(i)-|V|$ is nonnegative, otherwise the answer to k-$(G,L,\leqslant p)$ is negative. Let $G'=(V\cup\{x_1,\ldots,x_t\},E)$, where $x_1,\ldots,x_t\notin V$. If $t=0$, then no vertex is added to V. Let L' be such that $L'(x)=L(x)$ for every vertex $x\in V$ and $L'(y)=\{1,\ldots,k\}$ for every vertex $y\in\{x_1,\ldots,x_t\}$. We introduced no new colors and added $t\leqslant(k-1)|V|$ vertices. Moreover, by definition of t, there is a solution to the k-$(G,L,\leqslant p)$ problem if and only if there is a solution to the k-(G',L',p) problem. Thus, k-$(G,L,\leqslant p)$ polynomially reduces to k-(G',L',p) □

Lemma 2. *k-(G,L) polynomially reduces to k-(G,L,p).*

Proof. By Lemma 1, it is sufficient to reduce k-(G,L) to k-$(G,L,\leqslant p)$. Let $G=(V,E)$ and L constitute an instance of k-(G,L). We define a color-mapping p such that $p(i)=|V|$, $i\in\{1,\ldots,k\}$. To complete the proof, it is sufficient to see that there exists a solution to k-(G,L) if and only if there exists a solution to k-$(G,L,\leqslant p)$. □

Note that the reductions in these two proofs are polynomial in k and n. Since the input size of an instance of (G,L) is also polynomial in k and n (for each vertex of G, the list of no more than k colors must be given), we have also the following result:

Lemma 3. *(G,L) reduces to (G,L,p).* □

In order to prove that 3-(G,L,p) is $\mathcal{NP}$-complete for planar bipartite graphs, we will use the following theorem:

Theorem 1 (Kratochv'il [18])**.** 3-(G,L) is *$\mathcal{NP}$-complete for planar bipartite G.*

First observe that the reductions used in the proofs of Lemmas 1 and 2 applied to a planar graph preserve its planarity. Thus, by Lemmas 1 and 2, we have the following corollary of Theorem 1:

Corollary 1. 3-(G,L,p) *and* 3-$(G,L,\leqslant p)$ *are $\mathcal{NP}$-complete for planar bipartite G.* □

2.2. Complexity of 2-(G,L,p)

It is easy to see that 2-(G,L) is polynomially solvable. The case of 2-(G,L,p) is less straightforward, but the following theorem gives a similar result.

Theorem 2.2. 2-(G,L,p) *is polynomially solvable.*

Proof. First, observe that if G is not a bipartite graph then the answer to 2-(G,L,p) is 'no'. Note that it is a polynomial task to determine if G is bipartite, and to give a 2-coloring of a bipartite graph. Now, assume that G is bipartite. We claim that:

$$\text{We may assume that } L(v)=\{1,2\} \quad \forall v\in V(G). \tag{1}$$

Otherwise, let us assume that there exists a vertex v such that $L(v)=\{1\}$. Then, set $G'=G-\{v\}$, $p'(1)=p(1)-1$, $p'(2)=p(2)$, and

$$L'(u)=\begin{cases} L(u)-\{1\} & \text{if } u \text{ is a neighbor of } v,\\ L(u) & \text{otherwise.}\end{cases}$$

There exists an L-coloring S_1,S_2 of G such that $|S_j|=p(j)$, $j=1,2$ if and only if there exists an L'-coloring S_1',S_2' of G' such that $|S_j'|=p'(j)$, $j=1,2$. Now if we repeat this process, we finally obtain either an L-coloring of G, or an obstruction to an L-coloring of G, or a graph G' such that $L(v)=\{1,2\}$ $\forall v\in V(G')$. Let $C_1,\ldots,C_t$ be the $t\geqslant 1$ connected components of G. Let A,B be a 2-coloring of G, and $A_i=A\cap C_i$ and $B_i=B\cap C_i$ for all $i\in\{1,\ldots,t\}$. Without loss of generality, we may assume that for every $i=1,\ldots,t$, $a_i=|A_i|\geqslant|B_i|=b_i$. Let $v_i\in A_i$ with $i\in\{1,\ldots,t\}$ be arbitrarily chosen, and from now on fixed, vertices. For each connected component i, the only two possible colorings with two colors 1 and 2 consist in either coloring A_i with color 1 and B_i with color 2, or coloring A_i with color 2 and B_i with color 1. Define variable x_i

$$x_i=\begin{cases} 1 & \text{if } v_i \text{ is colored with 1,}\\ 0 & \text{otherwise.}\end{cases}$$

Thus, 2-(G,L,p) reduces to solving the following equation

$$\sum_{j=1}^{t}(a_ix_i+b_i(1-x_i))=p(1) \tag{2}$$

or

$$\sum_{j=1}^{t}(2a_i-|C_i|)x_i=p(1)+\sum_{j=1}^{t}a_i-n. \tag{3}$$

By our assumption, $2a_i-|C_i|\geqslant 0$ for all $i\in\{1,\ldots,t\}$, and therefore solving this equation is a special case of the subset sum problem, which can be solved by a dynamic programming algorithm in time $O(t\sum_{j=1}^{t}(2a_i-|C_i|))\subseteq O(tn)$ [11]. □

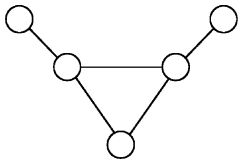

Fig. 1. The bull.

3. Treed graphs

In this section, we shall investigate the class of forests $\mathscr{F}$ and its closure under substitution $\mathscr{F}^*$. The graphs in $\mathscr{F}^*$ will be called *treed* graphs. We shall show that the problems k-(G,L), and k-(G,L,p) are solvable in polynomial time for treed graphs. This new class of graphs includes all P_4-free graphs, which will be shown in Theorem 4. Furthermore, since all chains P_n are treed graphs this class includes more than just the P_4-free graphs. Therefore, we extend the frontier of polynomial solvability of the problems k-(G,L) and k-(G,L,p) beyond the P_4-free graphs. Kobler shows in [17] that the treed graphs are weakly triangulated graphs (i.e., they do not contain any chordless cycle of length at least 4, or the complement of such a cycle as induced subgraph). Moreover, the class of treed graphs is distinct from the class of P_4-sparse graphs, defined as the graphs in which every set of five vertices induces at most one P_4 [15]. Indeed, P_n for $n \geqslant 5$ is a treed graph but not a P_4-sparse graph, and the bull (Fig. 1) is a P_4-sparse graph but not a treed graph.

To introduce treed graphs we need to define the concepts of graph substitution and closure under substitution.

Definition 1 (*Lovász* [19]). Substituting a graph G_2 for a vertex v of a graph G_1, denoted by $S(G_1,v,G_2)$, consists in taking the disjoint union of $G_1 - v$ and G_2, and adding an edge between each vertex of G_2 and each vertex of G_1 that is a neighbor of v in G_1.

Let $\mathscr{C}$ be a class of graphs; the closure $\mathscr{C}^*$ of $\mathscr{C}$ under substitution is the class of graphs obtained from graphs in $\mathscr{C}$ by repeated substitution by graphs also in $\mathscr{C}$. We shall show an algorithm to recognize graphs in $\mathscr{C}^*$ in polynomial time for any *nice* class $\mathscr{C}$ of graphs. The class $\mathscr{C}$ is called *nice* [7] if one can certify in polynomial time whether $G \in \mathscr{C}$, and each induced subgraph H of G is in $\mathscr{C}$. Obviously $\mathscr{F}$ is nice, thus our algorithm will also show how to recognize treed graphs in polynomial time. Our approach uses the following concept of modules.

Definition 2 (*Cournier and Habib* [6]). Consider a graph $G = (V,E)$, and a set of vertices D. The set D is a module if each vertex in $V \setminus D$ is adjacent either to all vertices in D or to none.

A module D is said to be trivial if $|D| \leqslant 1$ or $|D| = |V|$.

A graph is prime if it contains only trivial modules.

A sequence $S(\ldots S(S(H,u_0,S^*_{u_0}),u_1,S^*_{u_1})\ldots,u_t,S^*_{u_t})$ of substitutions for a nonprime graph G is called a *normal* sequence of substitutions for G if H is prime, $u_i \in V(H)$, and $S^*_{u_i}$ is either a normal sequence of substitutions for some graph or a prime graph with at least 2 vertices, $i=0,\ldots,t$. A normal sequence of substitutions for a prime graph G is G itself. We have the following lemma.

Lemma 4. *There exists a normal sequence of substitutions for any G.*

Proof. The proof proceeds by induction on the number of substitutions in a sequence of substitutions for G. It uses the following two simple properties of the substitution: Let G_1, G_2 and G_3 be three graphs and v a vertex of G_1. If w is another vertex of G_1, we have $S(S(G_1,v,G_2),w,G_3)=S(S(G_1,w,G_3),v,G_2)$. If x is a vertex of G_2, we have $S(G_1,v,S(G_2,x,G_3))=S(S(G_1,v,G_2),x,G_3)$. We shall omit details of the proof. □

The following algorithm calculates a normal sequence of substitutions for a graph G.

Function NORMALFORM (**INPUT**: **a graph** $G=(V,E)$) : $\rightarrow$ a normal sequence of substitutions for G.

1. Find a nontrivial module H (i.e. $|V|-1 \geqslant |H| \geqslant 2$) of G.
2. If there is no such a module then **return** G.
3. **Return** $S(\text{NORMALFORM}(G'),\ v_H,\ \text{NORMALFORM}(G[H]))$ where $G'=(V',E')$ with $V'=(V\setminus H)\cup\{v_H\}$ and $E'=(E(V\setminus H))\cup\{v_Hu \mid u\in N_G(H)\setminus H\}$.

A nontrivial module can be found in $O(|V|+|E|)$ time, [6]. Thus the time complexity of NORMALFORM is $O(|V|(|V|+|E|))$.

Any normal sequence of substitutions for a graph G can be used to decide in polynomial time whether G is in $\mathscr{C}^*$ for nice $\mathscr{C}$. The decision algorithm works as follows:

Function NICE (**INPUT**: **a graph** $G=(V,E)$): $\rightarrow$ 'yes', if $G\in\mathscr{C}^*$ else 'no'.

1. Set $S=$ NORMALFORM (G).
2. If all prime graphs in S belong to $\mathscr{C}$ then **r**eturn 'yes', else **r**eturn 'no'.

By definition of a nice class of graphs, $G\in\mathscr{C}$ implies that all induced subgraphs of G are also in $\mathscr{C}$. Thus, by induction and definition of substitution, $G\in\mathscr{C}^*$ implies that all induced subgraphs of G are also in $\mathscr{C}^*$. This shows that NICE is correct.

The complexity of NICE is $O(|V|(|V|+|E|)+|V|f(G))$ where $f(.)$ is the complexity of a certificate for $\mathscr{C}$. It is worth noticing that Kobler [17] proposes an $O(|V|+|E|)$ time algorithm to recognize treed graphs.

We are now ready to show how to compute the set k-$FCM(G,L)$ of all *feasible color-mappings* of a treed graph G. A function $q:\{1,\ldots,k\}\rightarrow\{0,\ldots,n\}$ is a feasible color-mapping of G if there exists a solution to k-(G,L,q). For convenience we shall represent q by an equivalent vector $(q(1),\ldots,q(k))$ in k-(G,L,q). By definition, the

cardinality of the set of all feasible color-mappings of G, denoted by k-$FCM(G,L)$, is less than or equal to $(n+1)^{k-1}$.

Theorem 3. *The set k-$FCM(G,L)$ can be computed in polynomial time for any treed graph G and fixed k.*

Proof. Our algorithm will use a *relaxed* normal sequence of substitutions for a treed graph $G=(V,E)$ as its input. This sequence is defined as follows. Let $s=S(\ldots S(S(H,u_0, S^*_{u_0}),\ u_1,S^*_{u_1}),\ldots,u_t,\ S^*_{u_t})$ be a sequence of substitutions for $G=(V,E)$, s is called a *relaxed* normal sequence of substitutions for $G=(V,E)$ if the following four conditions are fulfilled:

- H is a (not necessarily prime) forest;
- u_i is in $V(H)$, $i=0,\ldots,t$;
- $S^*_{u_i}$ is either a relaxed normal sequence of substitutions for some graph or a (not necessarily prime) forest with at least 2 vertices, $i=0,\ldots,t$;
- each prime graph of s is a simple vertex.

It is easy to turn any normal sequence of substitutions s into a relaxed one, for if there is a vertex v of some forest H in s which is not substituted, then we can replace H by $S(H',v',v)$ in s, where H' is H with v renamed to v'. Thus, after at most $|V|$ steps we obtain a relaxed normal sequence of substitutions for G. Now, let $s=S(S(\ldots S(H,a_0,S^*_{a_0}),a_1,S^*_{a_1}),\ldots,a_r,S^*_{a_r})$, be a relaxed normal sequence of substitutions for a treed graph $G=(V,E)$. We represent by $|s|$ the number of substitutions in s.

A *selector* R for H is a subset of $V(H)$ with exactly one vertex from each connected component of H. Consider a vector $W=(\alpha_1,\ldots,\alpha_k,q_1,\ldots,q_k)\in\mathbb{N}^{2k}$. Let $W^-=(\alpha_1,\ldots,\alpha_k)$ and $W^+=(q_1,\ldots,q_k)$ for W. W is *feasible* for (G,R) if there exists a solution to k-(G,L,W^+) with α_h vertices of the graph $\bigcup_{u\in R} S^*_u$ colored with $h, h\in\{1,\ldots,k\}$.

The set of all feasible vectors for (G,R), denoted by $F(G,R)$, includes no more than $(n+1)^{2k-2}$ elements.

Let G_1 and G_2 be two disjoint treed graphs with their extended normal sequences being $S_1=S(\ldots S(H_1,c_0,S^*_{c_0}),\ldots,\ c_p,S^*_{c_p}),\ldots)$ and $S_2=S(\ldots S(H_2,b_0,S^*_{b_0}),\ldots,b_s,S^*_{b_s})\ldots)$ respectively. Assume that G_1 is connected. Let $\{u\}$ and R_2 be selectors for H_1 and H_2, respectively.

Define $F(G_1,\{u\})\triangle F(G_2,R_2)=\{W_1+W_2\mid W_1\in F(G_1,\{u\})$ and $W_2\in F(G_2,R_2)\}$ and $F(G_1,\{u\})\perp F(G_2,R_2)=\{(W_1^-,W_1^+ + W_2^+)\mid W_1\in F(G_1,\{u\}), W_2\in F(G_2,R_2)$, and W_1^- and W_2^- are orthogonal$\}$.

We are ready now to define the function for calculating $F(G,R)$:

Function F (**INPUT: A treed graph** $G=S(\ldots(H,a_0,S^*_{a_0}),\ldots,a_r,S^*_{a_r})\ldots)$, where $S(\ldots(H,a_0,S^*_{a_0}),\ldots,a_r,S^*_{a_r})\ldots)$ is a relaxed normal sequence of substitutions. **A** selector R for H.$):\rightarrow F(G,R)$.

1. If G has only one vertex u, then **return**
$$\bigcup_{j \in L(u)} \{(0,\ldots,\alpha_j = 1,\ldots,0,\ldots,q_j = 1,\ldots,0)\}.$$
2. If $R = \{u\}$ and $G = S(u,u,S_u^*)$, then let $S_u^* = S(\ldots S(H_1,d_1,S_{d_1}^*),\ldots),d_q,S_{d_q}^*)\ldots)$ and let R_1 be a selector for H_1. **Return** $\{(W^+,W^+) \mid W \in F(S_u^*,R_1)\}$.
3. If $R = \{u\}$, then let $u_1,\ldots,u_t$ be all neighbours of u in H. Set $G_1 = S(u,u,S_u^*)$ and $G_2 = S(\ldots S(H-u,b_1,S_{b_1}^*),\ldots),b_s,S_{b_s}^*)\ldots)$, where $\{b_i\}_{i=1,\ldots,s} = V(H-u)$. **Return** $F(G_1,\{u\}) \perp F(G_2,\{u_1,\ldots,u_t\})$.
4. If $R = \{u,v,\ldots\}$, then set $G_1 = S(\ldots S(H_u,b_1,S_{b_1}^*),\ldots),b_s,S_{b_s}^*)\ldots)$, where H_u is the connected component of H containing u, $\{b_i\}_{i=1,\ldots,s} = V(H_u)$, and $G_2 = S(\ldots S(H - H_u,c_1,S_{c_1}^*),\ldots),c_p,S_{c_p}^*)\ldots)$, where $\{c_i\}_{i=1,\ldots,p} = V(H-H_u)$. **Return** $F(G_1,\{u\}) \triangle F(G_2,R\backslash\{u\})$.

Notice that in Steps 2–4, we do not have to find a relaxed normal sequence for the subgraphs of G, since they are given directly by the sequence for G.

The following two claims will prove that this function correctly constructs $F(G,R)$. The correctness of Step 2 derives directly from the definitions of a feasible vector and of $F(G,R)$.

Claim 1. *In Step* 3, $F(G,\{u\}) = F(G_1,\{u\}) \perp F(G_2,\{u_1,\ldots,u_t\})$.

Proof. Let $W \in F(G,\{u\})$. Then, there is a solution c to $k-(G,L,W^+)$ with W_h^- vertices in subgraph S_u^* of G colored with h. Define $a(v) = c(v)$ for v in G_1 and $b(v) = c(v)$ for v in G_2. Notice that $G_1 = S_u^*$. Colorings a and b uniquely define vectors $W_a = (W_a^-,W_a^+) \in F(G_1,\{u\})$ and $W_b = (W_b^-,W_b^+) \in F(G_2,\{u_1,\ldots,u_t\})$, respectively, such that $W_a^+ + W_b^+ = W^+$. Moreover, the sets of colors used in S_u^* and in $\bigcup_{v \in \{u_1,\ldots,u_t\}} S_v^*$ are disjoint. Thus, W_a^- and W_b^- are orthogonal. Therefore, $W \in F(G_1,\{u\}) \perp F(G_2,\{u_1,\ldots,u_t\})$.

Let $W \in F(G_1,\{u\}) \perp F(G_2,\{u_1,\ldots,u_t\})$. Then, there is a solution a to $k-(G_1,L,W_a^+)$ with $W_{ah}^- = W_{ah}^+$ vertices in subgraph S_u^* of G colored with h, and there is a solution b to $k-(G_2,L,W_b^+)$ with W_{bh}^- vertices in subgraph $\bigcup_{v \in \{u_1,\ldots,u_t\}} S_v^*$ of G colored with h. Moreover, $W = (W_a^-,W_a^+ + W_b^+)$, and W_a^- and W_b^- are orthogonal.

Define coloring c of G as follows
$$c(v) = \begin{cases} a(v) \text{ if } v \text{ is in } G_1, \\ b(v) \text{ otherwise.} \end{cases}$$
We observe that any vertex in S_u^* is adjacent to all vertices in $\bigcup_{v \in \{u_1,\ldots,u_t\}} S_v^*$ only. Therefore, coloring c is feasible for $F(G,\{u\})$, and consequently $W \in F(G,\{u\})$. □

Claim 2. *In Step* 4, $F(G,R) = F(G_1,\{u\}) \triangle F(G_2,R \setminus \{u\})$.

Proof. Let $W \in F(G,R)$. Then, there is a solution c to $k-(G,L,W^+)$ with W_h^- vertices in subgraph $\bigcup_{v \in R} S_v^*$ of G colored with h. Define $a(v) = c(v)$ for v in G_1 and $b(v) = c(v)$ for v in G_2. Colorings a and b uniquely define vectors $W_a = (W_a^-,W_a^+) \in$

$F(G_1,\{u\})$ and $W_b=(W_b^-,W_b^+)\in F(G_2,R\backslash\{u\})$, respectively, such that $W_a^+ + W_b^+ = W^+$. Thus, $W\in F(G_1,\{u\})\triangle F(G_2,R\setminus\{u\})$.

Let $W\in F(G_1,\{u\})\triangle F(G_2,R\setminus\{u\})$ Then, there is a solution a to $k-(G_1,L,W_a^+)$ with W_{ah}^- vertices in subgraph S_u^* of G colored with h, and there is a solution b to $k-(G_b,L,W_b^+)$ with W_{bh}^- vertices in subgraph $\bigcup_{v\in R\setminus\{u\}} S_v^*$ of G colored with h, such that W_a^- and W_b^- are orthogonal, and $W=(W_a^- + W_b^-, W_a^+ + W_b^+)$. Define coloring c of G as follows

$$c(v)=\begin{cases} a(v) \text{ if } v \text{ is in } G_1,\\ b(v) \text{ otherwise.}\end{cases}$$

Coloring c is feasible for G since no vertices in G_1 and G_2 are adjacent. Thus, $W\in F(G,R)$. □

We can now easily construct the set $k-FCM(G,L)$ using this function. Indeed, for an arbitrary selector R of G, we have $U\in k-FCM(G,L)$ if and only if there exists a vector $W\in F(G,R)$ with $W^+=U$.

Complexity analysis: We will prove that our algorithm runs in $O(n^{4k-3})$ time.

Proof.

(i) By definition of $\triangle$ operator, there exists a constant K_1 such that $F_1\triangle F_2$ can be computed in $K_1N_1N_2$ steps, where $N_i=|F_i|$ for $i=1,2$. Since $|F_i|\leqslant(n_i+1)^{2k-2}$, then $F_1\triangle F_2$ can be computed in $K_1(n_1+1)^{2k-2}(n_2+1)^{2k-2}$ steps.

(ii) By definition of $\perp$ operator, there exists a constant K_2 such that $F_1\perp F_2$ can be computed in $K_2(n_1+1)^{2k-2}(n_2+1)^{2k-2}$ steps.

(iii) Finding a connected component of a graph G can be done in K_3m steps where m is the number of edges of G and K_3 is a constant. So for a forest, it can be done in K_3n.

Let $K=\mathrm{Max}\{K_1,K_2,K_3\}$. We will prove by induction on n that F needs at most $K(|s|+1)(2n)^{4k-4}$ steps, where $|s|$ is the number of substitutions in the relaxed normal sequence of substitutions given in the input.

If G has only one vertex, only Step 1 is performed, which can be done in $|L(v)|$ steps. Thus, we may assume that $n>1$; one of the following three cases occurs:

Step 2: Let $s=S(u,u,S_u^*)$ and $s'=S_u^*=S(\ldots S(H_1,d_1,S_{d_1}^*),\ldots),d_q,S_{d_q}^*)\ldots)$. Then clearly, $|s'|=|s|-1$. By induction hypothesis, $F(S_u^*,R_1)$ can be computed in $K(|s'|+1)((2n)^{4k-4})$ steps. Since $|F(S_u^*,R_1)|\leqslant(n+1)^{2k-2}$, this case can be done in no more than

$$K((|s'|+1)((2n)^{4k-4})+(n+1)^{2k-2})$$
$$\leqslant K(|s'|+2)((2n)^{4k-4})=K(|s|+1)((2n)^{4k-4})$$

steps.

Step 3: By induction hypothesis, $F_1=F(G_1,u)$ and $F_2=F(G_2,\{u_1,\ldots,u_t\})$ can be computed in $K(|s_1|+1)(2n_1)^{4k-4}+(|s|-|s_1|+1)(2(n-n_1))^{4k-4}$ steps (where

$n_1 = |V(G_1)|$ and $s_1 = S(u,u,S_u^*)$). By (ii), since $n_1 \geqslant 1$ and $n - n_1 \geqslant 1$, $F_1 \perp F_2$ can be computed in no more than

$$\begin{aligned}
&K((|s_1|+1)(2n_1)^{4k-4} + (|s| - |s_1| + 1)(2(n-n_1))^{4k-4} \\
&\quad + (n_1+1)^{2k-2}(n-n_1+1)^{2k-2}) \\
&\qquad \leqslant K(|s|+1)((2n_1)^{4k-4} + (2(n-n_1))^{4k-4} + (2n_1)^{2k-2}(2(n-n_1))^{2k-2}) \\
&\qquad \leqslant K(|s|+1)(2(n_1+n-n_1))^{4k-4}
\end{aligned}$$

steps.

Step 4: By induction hypothesis, $F_1 = F(G_1, \{u\})$ and $F_2 = F(G_2, R \setminus \{u\})$ can be computed in $K(|s_1|+1)(2n_1)^{4k-4} + K(|s| - |s_1| + 1)(2(n-n_1))^{4k-4})$ steps (where $n_1 = |V(G_1)|$ and $s_1 = S(\ldots S(H_u, b_1, S_{b_1}^*), \ldots), b_s, S_{b_s}^*)\ldots)$). By (i) and (iii), since $n_1 \geqslant 1$ and $n - n_1 \geqslant 1$, $F_1 \perp F_2$ can be computed in no more than

$$\begin{aligned}
&K(n + (|s_1|+1)(2n_1)^{4k-4} + (|s| - |s_1| + 1)(2(n-n_1))^{4k-4} \\
&\quad + (n_1+1)^{2k-2}(n-n_1+1)^{2k-2}) \\
&\qquad \leqslant K(|s|+1)(n + (2n_1)^{4k-4} + (2(n-n_1))^{4k-4} + (2n_1)^{2k-2}(2(n-n_1))^{2k-2}) \\
&\qquad = K(|s|+1)(n + (2(n_1+n-n_1))^{4k-4} - (2n_1)^{2k-2}(2(n-n_1))^{2k-2}) \\
&\qquad < K(|s|+1)(2(n_1+n-n_1))^{4k-4}
\end{aligned}$$

steps.

Since we showed that the calculation of F needs at most $K(|s|+1)(2n)^{4k-4}$ steps for any relaxed normal sequence of substitutions s of G, we can use the one based on the normal sequence of substitutions obtained by NORMALFORM. In this case, $|s| \leqslant n$, and $F(G,R)$ is obtained in at most $K(2n)^{4k-3}$ steps.

The construction of $k - FCM(G,L)$ on the basis of $F(G,R)$, and the determination of an extended normal sequence of G can also be done in $O(n^{4k-3})$ time. □

Theorem 4. *P_4-free graphs are treed graphs.*

Proof. Let $G = (U,A)$ be a P_4-free graph. We prove by induction on $|U|$ that G is a treed graph. If $|U| \leqslant 2$, then the theorem holds. Now, assume that $|U| \geqslant 3$.

It is well-known [21] that if G is a P_4-free graph, then either G is not connected or $\bar{G}$ is not connected. If G is not connected, then we can apply the induction hypothesis to each of its connected components; and hence G is a treed graph. If G is connected, let $\bar{H}_1$ and $\bar{H}_2$ be nonempty graphs such that $\bar{G}$ is disjoint union of $\bar{H}_1$ and $\bar{H}_2$. Then $G = S(S(T,u,H_1),v,H_2)$ where T is a simple edge (u,v). □

The following corollaries are immediate consequences of Theorem 3:

Corollary 2. *k-(G,L,p), k-$(G,L,\leqslant p)$ and k-(G,L) are polynomial for G being a treed graph.* □

By Theorem 4 and Corollary 2, we have the following corollary which extends a result in [16].

Corollary 3. *k-(G,L,p), k-$(G,L,\leqslant p)$ and k-(G,L) are polynomial for G being a P_4-free graph.* □

4. Partial κ-trees

In this section, we show that k-(G,L,p) is polynomial for partial κ-trees with fixed κ. A graph T is called a κ-tree if and only if it satisfies one of conditions (i) or (ii):

(i) T is the complete graph on κ vertices,

(ii) T has a vertex x such that the neighborhood of x induces a clique of size κ, and $T-x$ is a κ-tree.

A graph G is called a partial κ-tree if it is a subgraph (not necessarily induced) of a κ-tree.

The classes of treed graphs and partial κ-tree graphs, for any fixed κ, are not comparable. On one hand, a chordless cycle of length at least 5 is a partial 2-tree but not a treed graph. On the other, a graph with minimum degree δ is not a partial $(\delta-1)$-tree. Thus, the complete bipartite graph $K_{\kappa+1,\kappa+1}$ which is a treed graph (as it is P_4-free), is not a partial κ-tree.

We will now show that k-$FCM(G,L)$ can be constructed in polynomial time for the partial κ-tree G when k and κ are fixed. Our proof will use a decomposition technique developed in [3,16] for partial κ-trees.

For a partial κ-tree $G=(V,E)$, a *nice tree-decomposition* of width κ is a pair (T,χ) with $T=(I,F)$ being an oriented binary tree with root r, and $\chi=\{X_i \mid X_i \subseteq V,\ i\in I\}$ being such that:

- $\bigcup_{i\in I} X_i = V$;
- for each edge $\{v,w\}\in E$, there is an $i\in I$ with $v,w\in X_i$;
- $X_i\cap X_m \subseteq X_j$ for each triple $i,j,m\in I$ with j being on the path between i and m in T;
- for each $i\in I$, $|X_i|\leqslant \kappa+1$;
- each node i of T is of one of the following four types:
  - leaf node: i is a leaf of T and $|X_i|=1$;
  - introduce node: i has one child j and there is a $v\in V$ such that $X_i = X_j\cup\{v\}$;
  - forget node: i has one child j and there is a $v\in V$ such that $X_j = X_i\cup\{v\}$;
  - join node: i has two children j_1 and j_2, and $X_i = X_{j_1} = X_{j_2}$.

Such a tree-decomposition can be found in linear time (see for instance [3] for a review of complexity results) and is used to construct the graph G, and the set k-$FCM(G,L)$, in a bottom-up way.

Consider a nice tree-decomposition $(T=(I,F),\ \chi=\{X_i \mid X_i \subseteq V,\ i\in I\})$ of a partial κ-tree$G=(V,E)$. For each node $i\in I$, we can define a triple, called *terminal graph*, $G_i=(V_i,E_i,X_i)$, where (V_i,E_i) is the subgraph of G induced by $V_i=\{v \mid v\in X_j$ and j is i or a descendant of i in $T\}$. The vertices in X_i are called the *terminals* of G_i. Notice that, by the definition of a nice tree-decomposition, the only vertices in V_i that can have neighbours in $V\setminus V_i$ are those in X_i. We also have that the subgraph (V_r,E_r), where r is the root of T, is the graph G itself. The rank $r(i)$ of node $i\in I$ is defined as the length of the unique path in T going from r to i. This implies that the rank of a node is smaller than the rank of any of its children. The number $r_{\max}$ represents the maximum rank in T. For simplicity of notation we will not distinguish between the list L and its restrictions to subgraphs of G; the restriction of L to a subgraph G_i will also be denoted by L.

We will now see how the set k-$FCM(G,L)$ is constructed, with the help of auxiliary sets. For a node $i\in I$, consider a sequence $W=(X_{i1},\ldots,X_{ik};\ q_{i1},\ldots,q_{ik})$ with $X_{ih}\subseteq X_i$ and $q_{ih}\in\mathbb{N},\ h=1,\ldots,k$. Let $W^+=(q_{i1},\ldots,q_{ik})$ for W. The sequence W is called *feasible* for i if:

- $X_{i1}\cup\cdots\cup X_{ik}=X_i$;
- $X_{ih}\cap X_{ih'}=\emptyset,\ 1\leqslant h\neq h'\leqslant k$;
- there exists a solution c to $k-(G_i,L,W^+)$ with $X_{ih}\subseteq c^{-1}(h)$ for all $h\in\{1,\ldots,k\}$.

The set of all feasible sequences for i is represented by F_i. By definition, we have $|F_i|\leqslant k^{\kappa+1}(|V_i|+1)^{k-1}$. Notice that a function q belongs to k-$FCM(G,L)$ if and only if there exists a sequence $(X_{r1},\ldots,X_{rk};q(1),\ldots,q(k))$ in F_r. Hence, the goal is to construct F_r, which is done by constructing all the sets $F_i,\ i\in I$, beginning with the nodes of maximum rank and finishing with the root. Depending on the type of the node i, the following four cases occur:

Leaf node: Let v be the vertex in X_i. F_i is the set of sequences $(X_{i1},\ldots,X_{ik};\ q_{i1},\ldots,q_{ik})$ such that there exists $h'\in L(v)$ with

$$X_{ih'}=\{v\},\quad q_{ih'}=1,$$

and

$$X_{ih}=\emptyset,\quad q_{ih}=0\ \ \forall h\neq h'.$$

Introduce node: Let j be the child of i and $v\in V$ the vertex such that $X_i=X_j\cup\{v\}$. F_i is the set of sequences $(X_{i1},\ldots,X_{ik};\ q_{i1},\ldots,q_{ik})$ such that there exist $h'\in L(v)$ and $(X_{j1},\ldots,X_{jk};\ q_{j1},\ldots,q_{jk})\in F_j$ with

$$\forall w\in X_{jh'},\ \{v,w\}\quad\text{is not an edge of } G,$$

$$X_{ih'}=X_{jh'}\cup\{v\},\quad q_{ih'}=q_{jh'}+1$$

and

$$X_{ih}=X_{jh},\quad q_{ih}=q_{jh}\ \ \forall h\neq h'.$$

Notice that, by definition of a nice tree-decomposition, $v\notin V_j$.

Forget node: Let j be the child of i and $v \in V$ the vertex such that $X_j = X_i \cup \{v\}$. F_i is the set of sequences $(X_{i1},\ldots,X_{ik};\ q_{i1},\ldots,q_{ik})$ such that there exist $(X_{j1},\ldots,X_{jk};\ q_{j1},\ldots,q_{jk}) \in F_j$ with

$$X_{ih'} = X_{jh'} \setminus \{v\}, \quad q_{ih'} = q_{jh'},$$

and

$$X_{ih} = X_{jh}, \quad q_{ih} = q_{jh} \ \forall h \neq h',$$

where h' is such that $v \in X_{jh'}$ (h' exists, since $X_{j1} \cup \cdots \cup X_{jk} = X_j$).

Join node: Let j_1 and j_2 be the two children of i. F_i is the set of sequences $(X_{i1},\ldots,X_{ik};\ q_{i1},\ldots,q_{ik})$ such that there exist $(X_{j_1 1},\ldots,X_{j_1 k};\ q_{j_1 1},\ldots,q_{j_1 k}) \in F_{j_1}$ and $(X_{j_2 1},\ldots,X_{j_2 k};\ q_{j_2 1},\ldots,q_{j_2 k}) \in F_{j_2}$ with

$$X_{j_1 h} = X_{j_2 h} = X_{ih} \quad \forall h \in \{1,\ldots,k\},$$

and

$$q_{ih} = q_{j_1 h} + q_{j_2 h} - |X_{ih}| \quad \forall \in \{1,\ldots,k\}.$$

It is not difficult to see that these cases construct the sets F_i correctly. In particular, the previously mentioned fact that a vertex $v \in V \setminus V_i$ cannot be adjacent to a vertex in $V_i \setminus X_i$, is used for the cases 'introduce node' and 'join node'. Therefore, the following algorithm computes the set of feasible color-mappings k-$FCM(G,L)$:

Function PARTIAL κ-TREE (INPUT: a partial κ-tree G): $\rightarrow$ k-$FCM(G,L)$.

1. Determine a nice tree-decomposition (T,χ) of G.
2. For $t := r_{\max}$ downto 0 do
   2.1. For each node i in T of rank t, compute the set F_i.
3. **Return** $\{(q(1),\ldots,q(k)) \mid \exists (X_{r1},\ldots,X_{rk})$ with $(X_{r1},\ldots,X_{rk};\ q(1),\ldots,q(k)) \in F_r\}$.

We can now state:

Theorem 5. *For any fixed k and κ, the set k-$FCM(G,L)$ can be determined in polynomial time if G is a partial κ-tree.*

Proof. We have seen how this set can be constructed and have explained why this construction works. It remains to give its complexity. As mentioned before, a nice tree-decomposition can be found in linear time in the number of vertices n of G. The number of nodes in the tree is also linear. Since the size of the sets F_i is bounded by $k^{\kappa+1}(|V_i|+1)^{k-1}$, a set F_i can be computed in $O(k^{2\kappa+2}n^{2k-2}) = O(n^{2k-2})$, the 'join node' case being the most demanding. Therefore the global complexity is at most $O(n^{2k-1})$. □

The following corollary is an immediate consequence of Theorem 5:

Corollary 4. *k-(G,L,p), k-$(G,L,\leqslant p)$ and k-(G,L) are polynomial for G being a partial κ-tree with fixed κ.* □

By Theorem 5, we can determine a further class of graphs for which k-(G,L,p) is polynomially solvable:

Theorem 6. *k-(G,L,p), k-$(G,L,\leqslant p)$ and k-(G,L) are polynomial for G being a triangulated graph.*

Proof. The size ω of the maximum clique in a triangulated graph G can be found in polynomial time [12]. If $\omega > k$ then the problem has clearly no solution. Otherwise, $\omega \leqslant k$ and therefore G is a partial k-tree, for which the answer can be found in polynomial time by Theorem 5. In fact, G is even a partial $(\omega - 1)$-tree (see for instance [2]). □

5. Conclusion

We considered the problem k-(G,L,p) of list coloring with cardinality constraints and fixed number of colors. This problem is $\mathcal{NP}$-Complete even for planar bipartite graphs and $k = 3$. We showed some polynomially solvable cases of the problem. In particular, we proved that 2-(G,L,p) can be solved in polynomial time, and that the same holds for k-(G,L,p) for partial κ-trees with fixed κ, and for triangulated graphs. We also introduced a new class of graphs, called treed graphs. We showed some properties of these graphs, and proposed a polynomial time algorithm to recognize them. Finally, we proved that k-(G,L,p) is polynomially solvable for treed graphs. With the same techniques as used in this paper, we can also prove that k-(G,L,p) can be solved in polynomial time for the complements of treed graphs and the complements of partial κ-trees with fixed κ [17]. It would be interesting to investigate the closure of triangulated graphs under substitution, and see if the technique presented in this paper is capable of proving polynomiality of k-(G,L,p) for the graphs in this closure.

6. Uncited References

[4,5,14,20]

Acknowledgements

We thank two anonymous referees who helped us to improve the presentation of Section 3 and suggested to us the proof of Theorem 6.

References

[1] M. Biro, M. Hujter, Zs. Tuza, Precoloring extension I. Interval graphs, Discrete Math. 100 (1992) 267–279.

[2] H.L. Bodlaender, A tourist guide through treewidth, Acta Cybernet. 11 (1993) 1–21.
[3] H.L. Bodlaender, Treewidth: algorithmic techniques, results, in: I. Privara, P. Ruzicka (Eds.), Proceedings of the 22nd International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Vol. 1295, Springer, Berlin, 1997, pp. 29–36.
[4] H. Buer, R.H. Möhring, A fast algorithm for the decomposition of graphs and posets, Math. Oper. Res. 8 (1983) 170–184.
[5] D.G. Corneil, H. Lerchs, L. Stewart Burlingham, Complement reducible graphs, Discrete Appl. Math. 3 (1981) 164–174.
[6] A. Cournier, M. Habib, A new linear algorithm for modular decomposition, in: S. Tison (Ed.), Proceedings of the Colloquium on Trees in Algebra and Programming—CAAP'94, Lecture Notes in Computer Science, Vol. 787, Springer, Berlin, 1994, pp. 68–84.
[7] C. De Simone, On the vertex packing problem, Graphs Combin. 9 (1993) 19–30.
[8] M. Dror, G. Finke, S. Gravier, W. Kubiak, On the complexity of a restricted list coloring problem, Discrete Math. 195 (1999) 103–109.
[9] P. Erdős, A.L. Rubin, H. Taylor, Choosability in graphs, Proceedings of the West Coast Conference on Combinatorics, Graph Theory and Computing, Arcata, CA, Congr. Numer. XXVI (1979) 125–157.
[10] D.G. Fon-Der-Flaass, Arrays of distinct representatives—a very simple NP-complete problem, Discrete Math. 171 (1997) 295–298.
[11] M.R. Garey, D.S. Johnson, Computers and Intractability: a Guide to the Theory of NP-completeness, Freeman, New York, 1979.
[12] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[13] S. Gravier, Coloration et produits de graphes, Thesis, Université Joseph Fourier, Grenoble, France, 1996 (in French).
[14] M. Habib, M.C. Maurer, On the X-join decomposition for undirected graphs, Discrete Appl. Math. 1 (1979) 201–210.
[15] C. Hoàng, Thesis, McGill University, Montréal, Canada, 1985.
[16] K. Jansen, P. Scheffler, Generalized coloring for tree-like graphs, Discrete Appl. Math. 75 (1997) 135–155.
[17] D. Kobler, Modèles biologique en optimisation combinatoire et modèles mathématiques en génétique, Thesis, Swiss Federal Institute of Technology in Lausanne, Switzerland, 1999 (in French).
[18] J. Kratochv'il, Precoloring extension with fixed color bound, Acta Math. Univ. Comenian 62 (1993) 139–153.
[19] L. Lovàsz, Normal hypergraphs and the perfect graph conjecture, Discrete Math. 2 (1972) 253–267.
[20] R.M. McConnell, J. Spinrad, Linear-time modular decomposition and efficient transitive orientation of undirected graphs. Proceedings of the fifth Annual ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 536–545.
[21] S. Seinsche, On a property of the class of n-colorable graphs, J. Combin. Theory B 16 (1974) 191–193.
[22] V.G. Vizing, Vertex colorings with given colors, Metody Diskret. Analiz. 29 (1976) 3–10 (in Russian).
[23] D. de Werra, Restricted coloring models for timetabling, Discrete Math. 165/166 (1997) 161–170.